Chapter 1

INTRODUCTION

Voluminous data is being collected by various organizations today. This will continue. It becomes computationally inefficient to analyze such huge amount of data. Research in data mining has addressed problems in discovering knowledge from these continuously growing large data sets. The amount of raw data available to researchers, in a variety of scientific fields, has been increasing at an exponential rate.

The history of extraction of patterns from data is centuries old. An earlier method, which has been extensively seen, uses Bayes’ theorem (1700s) and regression analysis (1800s). (Pang-NingTan, M S V K 2006) In the field of computer technology, using the ever-growing power of computers, an essential tool for working with data, has been developed such that, it is possible to work with increasing sizes of the datasets and complexity. There is also an urgent need to further refine the automatic data processing, which has been assisted by other discoveries in computer science. This means that the ability for data collection, storage and manipulation of data has increased. Historically, the field of finding useful patterns in data has a myriad of names including, but not limited to, Data mining, Knowledge Extraction, Information discovery, data archaeology and data pattern processing. Statisticians’ use the term of Data mining and this is also very popular in the field of databases.

Data mining is the process of automatically discovering useful information in large data repositories. It is an induction based learning strategy that builds models to identify hidden patterns in data. A model created by a data mining algorithm is a conceptual generalization of data. The generalization may be in the form of tree, a network, an equation or a set of rules. Data mining is a multi-step process that requires access to and preparation of data for a data mining algorithm, mining the data, analyzing results and taking appropriate action. The data to be accessed can be stored in one or more operational databases, a data warehouse or a flat file.
Data is mined using supervised learning or unsupervised clustering. The most common type of data mining is supervised. It determines the classification of new, previously unclassified instances. With unsupervised clustering, predefined classes do not exist. Instead, data instances are grouped together based on a similarity scheme defined by the clustering model. Data mining is a one step advanced process that provides potentially useful information even when little or no idea is available about what we are looking for (Matthias Jarke et al. 1999).

In any organization, the quantum of data stored in database continues to grow fast. This large amount of stored data contains valuable hidden knowledge, which is used to improve the decision making process of the organization. Data about previous sales might contain interesting relationships between products and customers. The discovery of such relationships can be very useful to increase the sales of the company. Data mining derives its name from the similarities between searching for valuable information in a large database and mining rocks for a vein of valuable ore. Both imply either sifting a large amount of material or ingeniously probing the material to exactly pinpoint where the value resides.

In order to discover such types of relationship, data mining is used on datasets as a four-step process. The four steps (Devendra Kumar Tiwary, 2010) are:

i) Assemble data to enable application of data mining. These assembled data are kept in the database. There is no need to assemble huge amount of data for the data mining algorithm. Most of the data mining algorithms work best on a few datasets.

ii) Apply data mining tools on datasets. At this stage, many choices have to be made. The questions that arise are: Should learning be supervised or unsupervised, which instance of data is to be used to build model and which instance will test the model, which attributes will be selected from the list of available attributes.

iii) Interpretation and evaluation of results pertains to examining the output of data mining tool to determine the usefulness and interesting nature of the discovered result. If the result is not satisfactory, data mining step is repeated using new attributes or process is sent back to data warehouse for repeating data extraction process.
iv) Application of the result is the final step of the data mining process. The ultimate goal of data mining is to apply the result in new situations with changed attributes or instances. The result is applied in real life situations to solve problems.

Data mining techniques are deployed to clean up large databases in order to find novel and useful patterns that might otherwise remain unknown. Linking records from several databases or finding duplicate records in one data set have become increasingly important tasks in the data preparation phase of many data mining projects. The aim of such linkages is to merge all records relating to the same entity, such as a patient, a customer, or a business. As common unique entity identifiers (or keys) are often not available, the linkage process needs to be based on the available data – likes names and addresses – and this becomes increasingly difficult with today's large data collections. Businesses routinely deduplicate and link their data sets to compile mailing lists, while in taxation offices and departments of social security, data linkage can be used to identify people who register for benefits multiple times or who work and collect unemployment money.

Record linkage, which is the task of matching equivalent records, differs syntactically and this was first explored in the late 1950s and 1960s. (Fellegi, I P & Sunter, A B 1969) studied record linkage in the context of matching population records. This provides a theoretical foundation for subsequent work on the problem. In addition the authors described several key insights that still lie at the base of many modern name-matching systems:

1. A Pair of records can be represented using a vector of features that describes the similarity between individual record fields. Features can be Boolean (for example, last-name-matches), discrete (for example, first-n-characters-of-name-agree), or continuous (for example, string-edit-distance-between-first-names)

2. The problem of identifying matching records can be viewed as a task of placing feature vectors for record pairs into three classes: links, non-links, and possible links. These correspond to equivalent, non-equivalent, and possibly equivalent (for example, requiring human review) record pairs, respectively.
3. A system can perform record-pair classification by calculating the ratio \( \frac{P(\gamma | M)}{P(\gamma | U)} \) for each candidate record pair, where \( \gamma \) is a feature vector for the pair and \( P(\gamma | M) \) and \( P(\gamma | U) \) are the probabilities of observing that feature vector for a matched and mismatched pair, respectively. Two thresholds based on desired error levels—\( T_\mu \) and \( T_\lambda \)—optimally separate the ratio values for equivalent, possibly equivalent, and non-equivalent record pairs.

4. When no training data in the form of duplicate and nonduplicate record pairs is available, name matching can be unsupervised, wherein conditional probabilities for feature values are estimated using field value frequencies.

5. Most record pairs are clearly nonduplicates, these record pairs need not be considered for matching. Blocking databases so that only records in blocks are compared significantly improves efficiency.

The first four insights lay the groundwork for accurate record pair matching, while the fifth provides for efficiently processing large databases.

Recognition of similarities in large collections of data is a major issue in the context of designing information integration systems. The wide exploitation of new techniques and systems for generating, collecting and storing data has made available a huge amount of information. Large quantities of such data are stored as continuous text, such as personal demographic data, bibliographic information, phone and mailing lists. Often, the integration of such data is a problematic process that involves dealing with two major issues, namely structural and syntactic heterogeneity. The former occurs when the data does not explicitly exhibit a common field structure.

In this case, schema reconciliation techniques (Agichtein, E & Ganti, V 2004; Monge, A E & Elkan, C P 2001; Cesario, E et al. 2008) allow the segmentation of textual sequences into a uniform schema. However, whether or not reconciled into a common database schema, data can still be affected by syntactic heterogeneity. This is a fundamental issue in the context of information integration systems. This consists of discovery of duplicates within the integrated data. Syntactically different records, as a matter of fact, refer to a same real-world attribute. De-duplication is necessary in several application settings. A typical example is the reconciliation of demographic data sources into a data warehousing setting. Names and addresses can be stored in rather
different formats, thus raising the need for an effective reconciliation strategy that could be crucial for effective decision-making.

In these cases, a (typically large) volume of small strings is analyzed to reconstruct the semantic information on the basis of little syntactic information. Let us consider, e.g., a banking scenario, where the main interest is to rank the credit risk of a customer, by looking at the past insolvency history. Since information about payments may come from different sources, each of which conforming to a possibly different encoding format for the data, de-duplicating demographic tuples is crucial in order to correctly analyze the attitude at insolvency of the customer.

In such application scenarios, tuples usually correspond to small sequences of strings, characterized by an inherent segmentation in specific semantic entities. However, such segmentation is not known in advance and this further exacerbates the difficulties behind the identification of duplicates.

With the increasing importance of record linkage in a variety of data-analysis applications, developing effective and efficient techniques for record linkage has emerged as an important problem (Loshin, D 2000). It is further reinforced by the emergence of numerous organizations (e.g., Trillium, FirstLogic, Ascential Software, DataFlux) that are developing specialized domain-specific record-linkage and data-cleansing tools.

Three primary challenges arise in the context of record linkage. First, it is important to determine similarity functions that can be used to link two records as duplicates (Cohen, W W et al. 2000; Winkler, W 1994). Such a similarity function consists of two levels. First, similarity metrics need to be defined at the level of each field to determine similarity of different values of the same field. Next, field-level similarity metrics need to be combined to determine the overall similarity between two records. At the field level, a typical choice is string edit distance, particularly if the primary sources of errors are typographic and the type of the data is string. However, as the above examples illustrate, domain-specific similarity measures (e.g., different functions for people’s names, addresses, paper references, etc.) are more relevant. At the record level, merging rules that combine field-level similarity measures into an overall record similarity need to be developed. Approaches based on binary classifiers, expectation maximization (EM) methods, and support vector machines have been proposed in the literature (Bilenko, M & Mooney, R J 2002; Cohen, W W et al. 2000; Winkler, W 1994).
The second challenge in record linkage is to provide user-friendly interactive tools for users to specify different transformations, and use the feedback to improve the data quality. Recently a few works (Elfeky, M G et al. 2002; Galhardas, H et al. 2001; Raman, V & Hellerstein, J M 2001; Sarawagi, S et al. 2002) describe how to solve this problem.

The third challenge is that of scale. A simple solution is to use a nested-loop approach to generate the Cartesian product of records, and then use the similarity function(s) to compute the distance between each pair of records. This approach is computationally prohibitive as the two data sizes become large. The scalability issue of record linkage has been studied (Hernandez, M A & Stolfo, S J 1995). Since the original work (Hernandez, M A & Stolfo, S J 1995), many new data-management techniques have been unearthed with a direct bearing on the record-linkage problem. In particular, techniques for mapping arbitrary similarity spaces into similarity/distance-preserving multidimensional Euclidean spaces (Bourgain, J 1985; Faloutsos, C & Lin, K-I 1995; Hristescu, G & Farach-Colton, M 1999; Kruskal, J B & Wish, M 1978; Shinohara, T et al. 2000; Wang, J T L 1999; Young, F W & Hamer, R M 1987) have been developed. Furthermore, many efficient multidimensional similarity joins have been studied (Alsabti, K et al. 1998; Brinkhoff, T et al. 1993; Corral, A et al. 2000; Koudas, N & Sevcik, K C 1997; Kukich, K 1992; Mamoulis, N & Papadias, D 1999; Sevcik, K C & Koudas, N 2000; Shim, K et al.1997; Shin, H et al. 2000).

Some of the issues of record linkage are illustrated with a straightforward example as in Table 1.1. The following three pairs represent three individuals. In the first two cases, a human being could generally determine that the pairs are the same. In both situations, the individuals have reasonably similar names, addresses, and ages. Software that automates the determination of match status is welcome. The third situation may reveal that the first record of the pair is a medical student at the university twenty years ago. The second record is from a current list of physicians in Detroit who are known to have attended the University of Michigan.

It is associated with a doctor in a different city known to have attended a medical school at the university. With good automatic methods, the first two pairs as representing the same person could be determined. With a combination of automatic methods and human understanding, we might determine that the third pair is the same person.
Table 1.1 Elementary Examples of Matching Pairs of Records

(Independent on Context)

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>John A Smith</td>
<td>16 Main Street</td>
<td>16</td>
</tr>
<tr>
<td>J H Smith</td>
<td>16 Main St</td>
<td>17</td>
</tr>
<tr>
<td>Javier Martinez</td>
<td>49 E Applecross Road</td>
<td>33</td>
</tr>
<tr>
<td>Haveir Marteneez</td>
<td>49 Applecross Road</td>
<td>36</td>
</tr>
<tr>
<td>Gillian Jones</td>
<td>645 Reading Ave</td>
<td>24</td>
</tr>
<tr>
<td>Jillian Brown</td>
<td>123 Norcross Blvd</td>
<td>43</td>
</tr>
</tbody>
</table>

Another application of current interest is the use of data linkage in crime and terror detection. Security agencies and crime investigators increasingly rely on the ability to quickly bring up files for a particular individual, which may help to prevent crimes or terror by early intervention. Sophisticated data linkage techniques need application if no unique identifier is shared by all of the data sets to be linked. These techniques can be broadly classified into deterministic or rules-based approaches (in which sets of often very complex rules are used to classify pairs of records as links, i.e. relating to the same entity, or as non-links), and probabilistic approaches (in which statistical models are used to classify record pairs). Probabilistic methods can be further divided into those based on the classical probabilistic record linkage theory as developed by (Fellegi & Sunter, 1969) and newer approaches using maximum entropy, clustering and other machine learning techniques (Cohen, W, 1998; Winkler, W E, 1999; Lee, M L et al. 2000; McCallum, A, et al. 2000; Elfeky, M G et al. 2002; Nahm, U Y et al 2002; Bilenko, M, Mooney, R J, 2003; Cohen, W W et al. 2003; Gu, L & Baxter, R, 2004; Yancey, W E, 2004).

Computer-assisted data linkage goes back as far as the 1950s, for which Fellegi and Sunter, 1969, have provided the theoretical foundation. The basic idea is to link records by comparing common attributes (or fields), which include person identifiers (like names, dates of birth, etc.) and demographic information (for example addresses). Instead of using only exact comparisons between attribute values, approximate string comparison
methods can be applied (Cohen, W W, et al. 2003; Porter, L & Winkler, W E, 1997) resulting in partial agreement values. While, in the past, clerical review was manageable in a reasonable amount of time, linking today’s large administrative data collections (practically with millions of records) make the clerical review process impossible, as tens or even hundreds of thousands of record pairs will be put aside for clerical review. Clearly, what are needed are more accurate and automated decision models that will reduce the amount of clerical review needed – if not it has to be eliminated – while keeping a high linkage quality.

The majority of solutions for record linkage treat it as a modular problem and consist of multiple stages. In the first stage, a function is selected for computing the similarity between all pairs of potentially co-referent records. This similarity function can either be hand-tuned or its parameters can be learned from training data. In the second stage, a blocking method is used to select a set of candidate record pairs to be investigated for co-reference, since it is typically prohibitively expensive to compute pair wise similarities between all pairs of records in a large database. In the final linkage stage, similarity is computed between candidate pairs, and highly similar records are identified and linked as describing the same entity. This can be achieved either via pair wise or via collective inference over individual record pairs.

Pair wise approaches classify all candidate record pairs into two categories: “matches” or “non-matches”, where each candidate pair is classified independently of others. Some of the methods that employ the pair wise approach include the EM-based technique for finding optimal entity matching rules (Winkler, W E 1988), the sorted neighbourhood method for limiting the number of potential candidate pairs (Hernandez, M A & Stolfo, S J 1995), and the domain-independent three-stage iterative merging algorithm for duplicate detection (Monge, A E & Elkan, C P 1997).

In contrast, collective linkage methods take a more global view instead of processing each candidate pair independently.

These methods consider multiple linkage decisions in conjunction, and perform simultaneous inference to find groups of co-referent entries that map to the same underlying entity. Recently proposed algorithms that belong to this category include context-sensitive duplicate inference by iterative processing of the data (Bhattacharya, I &
Getoor, L 2004), learning a declarative relational probability model that explicitly specifies the dependencies between related linkage decisions (Pasula, H et al. 2003), and multi-relational inference using conditional probabilistic models to simultaneously detect matches for all related candidate pairs (Parag & Domingos, P 2005; McCallum, A & Wellner, B 2005). While some of these collective approaches have been shown to be more accurate than the pair wise approach on certain domains, the simultaneous inference process makes these methods more computationally intensive.

In general, the record linkage process comprises of the following sub processes (Christen, P 1998) as shown in the Fig.1.1.

![Fig.1.1 The general process of matching two databases](image)

In this research, we have concentrated on the following sub processes in record linkage.

- Record pair comparison
- Classification
**Evaluation**

We have applied the sub processes to find matching records from web databases instead of traditional databases.

Web databases are those which dynamically generate Web pages in response to the user queries, which are available on the Internet. Web databases constitute an enormous repository of searchable data on an extremely diverse collection of subjects, useful for a wide range of Database Applications. They are the databases that produce the results dynamically in response to the user queries. Web databases contain different types of data such as text, images, documents, and links. Users can access the Web databases by submitting queries via the query interface. The data in the Web databases are often redundant i.e., the records in one database can also be found in other databases. The crucial task in Web database is to identify the duplicate records by analyzing and matching records from different data sources.

Most of the Web databases are accessible via query interfaces. Based on the query received, web server retrieves the corresponding result from the backend database and returns it to the user. For example, Fig.1.2 and Fig.1.3 show some of the query results returned by two online bookstores, namely, bookadda.com and infibeam.com respectively, in response to the same query “harry potter and” over the title field. It can be seen that the record numbered 2 in Fig.1.2 and record numbered 4 in Fig.1.3 both of them refer to the same book. In comparison, the record numbered 4 in Fig.1.2 and record numbered 3 in Fig.1.3, both of them refer to the same book if the interest lies only in the book title. Thus, identifying the duplicates (i.e. two or more records describing the same real world entity) is a problem which attracts much attention from various research fields, which include databases, data mining, Artificial Intelligence (AI), Natural Language Processing (NLP), etc. The records to be matched are highly query dependant in Web databases, because they can only be obtained through online queries. Hand-coding and offline-learning approaches are not appropriate due to non-availability of full datasets and no good representative data for training. Also, the rules learned on the representatives of a full data set may not work well on a biased part of that data set. A query for the books of a specific author is considered. The resulting records for this query will have a unique name for the author field.
1. **Harry Potter and the Philosopher's Stone** by J.K. Rowling
   - Price: Rs.399  Rs.280
   - Discount: Rs.119 (30%)
   - Delivered in 2-4 business days.
   - Language: English | Binding: Paperback

2. **Harry Potter and the Chamber of Secrets** by J.K. Rowling
   - Price: Rs.399  Rs.268
   - Discount: Rs.131 (33%)
   - Delivered in 2-4 business days.
   - Language: English | Binding: Paperback

3. **Harry Potter and the Prisoner of Azkaban** by J.K. Rowling
   - Price: Rs.399  Rs.250
   - Discount: Rs.149 (37%)
   - Delivered in 2-4 business days.
   - Language: English | Binding: Paperback

4. **Harry Potter and the Deathly Hallows** by J.K. Rowling
   - Price: Rs.599  Rs.377
   - Discount: Rs.222 (37%)
   - Delivered in 2-4 business days.
   - Language: English | Binding: Paperback

Fig.1.2. Query results from www.bookadda.com
1. **Harry Potter and the Philosopher's Stone** by J.K. Rowling (Paperback 2001)

₹ 651  
Buy This Item

26 new & used from sellers starting at ₹ 596

**Available. Ships Free to India in 15-18 days**

Larger print edition. An ancient Greek edition, translated by Andrew Wilson. Irish edition translated by Maire Nic Mhaolain. When a letter arrives for unhappy but ordinary Harry Potter, a decade-old secret is revealed to him that apparently he's the last to know. His parents were wizards, killed by a Dark...  
more


2. **Harry Potter and the Deathly Hallows** by J.K. Rowling (Paperback 2010)

₹ 649 (22% OFF)  
Buy This Item

42 new & used from sellers starting at ₹ 571

**Available. Ships Free to India in 15-18 days**

The Deluxe Edition includes an exclusive insert featuring near-scale reproductions of Mary GrandPré's interior art, as well as never-before-seen full-color frontispiece art on special paper. The custom-designed slipcase is full-stamped and contains a full-cloth case book that has been blind-stamped on front and back cover with foil stamping...  
more


3. **Harry Potter and the Deathly Hallows** by J.K. Rowling (Paperback 2010)

₹ 461 (23% OFF)  
Buy This Item

96 new & used from sellers starting at ₹ 569

**In Stock. Ships Free to India in 2-3 days**

About the Book: In this thrilling climax to the phenomenally bestselling series, J.K. Rowling reveals the fate of Harry Potter when he must confront the Dark Lord. Harry Potter is preparing to leave the Dursleys and Privet Drive for the last time. But the future that awaits him is...  
more


₹ 80 (20k off)

**Fig.1.3. Query results from www.infibeam.com**
Here the author field of these records is ineffective for distinguishing the records, which should be matched, and those that should not. In order to reduce the influence of such fields, their weights should be lowered than the other fields or even be zero. For new queries, depending upon the results returned, field weights should be adjusted, which makes the supervised learning methods less applicable. To overcome such problems, a new method called Unsupervised Conflation Method is proposed in this thesis. Highlights of this method are as follows:

1. Different fields need to be assigned different weights in a dynamic manner.
2. Use of sample data consisting of record pairs from different sources (for negative training set) and record pair from same data source.

Each field weight is set according to its relative distance – dissimilarity among records from the negative training set. With the help of the weights, first classifier matches the records from different data sources. With the matched records (positive set), and the non-duplicate records (negative set), second classifier identifies new duplicates. All the identified duplicates and non-duplicates are used to adjust the field weights set in the previous step and new iterations begins by the first classifier to identify new duplicates. When no new duplicates are identified, the iteration is terminated. An approximated negative training set is used for learning (which may contain some positive examples).